# JEAN-MARC GILLIOT QGIS-tuto.fr



la Baudrière



# INTRODUCTION A L'AUTOMATISATION DANS QGIS : MODE BATCH, MODELEUR GRAPHIQUE, SCRIPT PYTHON et R





# Version d'octobre 2024



Grande école européenne d'ingénieurs et de managers dans le domaine du vivant et de l'environnement



Jean-marc.gilliot@agroparistech

Conseil A vous de jouer A ttention Untée	objectif
clic souris : gauche droit VouTube tuto. Vidéo	à retenir
1. Démarrage de QGIS	
2. Chaine de traitements et traçabilité	
3. Historique de la boite à outils de traitement de QGIS	6
4. Mode Batch (processus de lot) de QGIS	88
4.1 Le menu « processus de lot » dans la boite à outils de traitements	9
4.2. Remplissage du tableau des traitements et nommage automatique	۲۲ 1 ع
5. Le modeleur graphique de chaine de traitements QGIS	13 1/
5.1. L'interface du modeleur graphique	14
5.2 Ajour des données : parametrisation des entrées	15 14
5.5 A jour des algorithmes (traitements)	10
5.4 Le nouveau modele dans la borre a ourris de traitement	19
5.5 Connecter les digorithmes entre eux : input <-> output	19 25
6 Scrint nython at R dans l'interface de AGIS	22 ۹C
61 Code dans la console python de QGTS et l'interface iface	30
6.2 Exemples avec iface dans un fichier script de la console	
6.2.1. Saisir et exécuter un fichier script dans la console	
622 Accéder aux couches objets et attributs en python avec iface	35
a) Accès aux attributs	35
b) Accès aux entités sélectionnées : selectedFeatures()	35
c) Boite de dialogue de type « input text »	
d). Accès à la géométrie	
e). Calculs dans un champ	
6.3. Code processing.run() dans la console python (Toolbox processing)	
6.4. Code dans un script Python dans la boite à outils Traitement : PyQGIS	44
6.4.1. Création d'un script comme dans la console	44
6.4.2. Les « import » python en début du script	44
6.4.3. processing.run() : les algorithmes de la boite de Traitement dans un script	:44
6.4.4. « décorateur » @alg : créer un script dans la boite de traitements :	45
6.4.5. Script R dans la Toolbox de traitement	47
7. Script python et R « standalone » en dehors de QGIS	
7.1. qgis_process : Code « processing » dans un script externe à QGIS	51
7.1.1 qgis_process en ligne de commande	52
7.1.2 qgis_process dans un script python « standalone »	54
7.1.3 ggis_process dans un script R « standalone »	57
7.2 programmes python externe à QGIS	59
8. Script python et R « standalone » en RMarkdown	60
8.1 Notion de fichier Markdown	60
8.2 Fichier RMarkdown	61





Lancer QGIS Desktop 3.34

https://qgis.org/downloads/

Sur le site de téléchargement de QGIS, privilégier la dernière version dite « LTR » (long time release) qui est la version la plus stable car elle subit des tests plus rigoureux avant sa sortie et reçoit au moins un an de mises à jour de correction de bogues.

Long Term Version for Windows (3.34 LTR)

			_
<b>Q</b> Projet sans titre — QGIS		- 0	×
Projet Éditer Vue Couche Préfére	ences Extensions <u>V</u> ecteur <u>R</u> aster <u>B</u> ase de d	onnées Internet Maillage Iraitement Aide	
🗋 📁 🗟 🔂 🕄 🕚	J R R 🤍 Q 🕱 🗸 🕂 🗬		
🧔 📽 Vi 🔏 🖏 🕅 刘	レノ 局・治 友・副 着 べ 自 昏	ㅎ ㅎ = 역 역 역 영 영 영 영 영 영 영 영	
🔣 + 🗐 + 🌄 + 🛶			
Explorateur	Actualités		
Marque-pages		Planned end of life for Windows 32-Bit Support	8 Î
<ul> <li>i Gaccueil</li> <li>i C</li> <li>i E</li> <li>i U</li> <li>i Z</li> <li>i GeoPackage</li> <li>✓ Spatialite</li> </ul>		QGIS will drop 32-bit support on Windows after the QGIS 3.16 release when we update our Qt dependencies to Qt 5.15. The update to Qt 5.15 is an important step towards staying in sync with Qt developments. Qt 5.15 is the minimum version that will provide forward compatibility with Qt 6. By updating to 5.15, we, therefore, ensure that QGIS is future proof. <b>Please double-click this entry to find</b> <b>out more.</b>	2
PostGIS MSSQL Oracle DB2 WMS/WMTS HVector Tiles	Modèles de projet	QGIS Monthly Virtual Fridays! Have you been missing all the buzz and excitement around the 6-monthly in-person meetups we used to hold? Perhaps you live far away from Europe and could never join these meetups in person? Well we have great news for you! Starting August Friday 28th 2020.	D -
Couches			
≪ 48, ∞, ₹ %, ~ 10, 12 🖬 🗔		Nouveau projet vide	
		EPSG:4326 - WGS 84	
Q Tange pour trouver (Ctrl-H)	Prêt Coord	nondo 🔍 Erhalla 13717264 y 🔎 Lauron Tonos. 🔿 Dotation 10.0 🔿 🖓 √ Bendu 🐗 EDDrugges 🕷	
Let eater boar a gare (earrie)	Cond		Κ.

Télécharger le jeu de données « La Flèche » = fichier La\_Fleche.zip

Décompresser le fichier ZIP dans votre disque de travail

Le dossier 📙 LA_FLECH	IE contient les données du TD
Depuis QGIS ouvrir le ficht qui est à la racine du dossie	ier de projet <b>IA_FLECHE.qgs</b> er « LA_FLECHE »
Menu $ ightarrow$ Projet $ ightarrow$ Ouvrir	Ou le bouton ouvrir 📁
C	🞗 Projet sans titre - QGIS

On peut aussi simplement double cliquer sur la\_fleche.qgs dans l'explorateur de fichiers Windows, pour lancer QGIS avec le projet, ou depuis le panneau explorateur de fichier QGIS

Projet <u>É</u>diter V<u>u</u>e <u>C</u>ouche <u>P</u>références



Le plus souvent pour réaliser une analyse géographique sous SIG, on va enchainer une série d'opérations avec leurs entrées et leurs paramètres, on parle de « chaine de traitements » comme dans la figure ci-dessous :

🕀 zones	💥 🔶 🗣 parcelles	8
	• Entrée +	
	🚿 Intersection 🧮	
	Sortie -	
	Intersection: Intersection	
	Entrée +	
	* Calculatrice de champ	83
	Sortie + •	
	⇔ sortie	83

On peut être amené à reproduire une même chaine de traitements, par exemple avec des données différentes, il est alors très difficile de garantir qu'on aura appliqué les mêmes paramètres que la première fois, pour pouvoir comparer les résultats obtenus avec des données différentes.

Il y a donc nécessité de garder la trace détaillée de toutes les opérations et de leurs paramètres afin de pouvoir les reproduire à l'identique : on parle de traçabilité. On peut noter dans un fichier texte toutes ces informations mais cela est très fastidieux.

Implémenter une chaine de traitement sous la forme d'un code informatique est une réponse efficace au besoin de traçabilité.

En effet on peut alors facilement réexécuter à l'identique un même code et donc reproduire à l'identique une chaine de traitement. Le code servira aussi de documentation pour conserver la mémoire de ce qui a été fait.

Maupar la la Guillardere bla tradioni fiere a la la Robiniere la Burochore a la la Robiniere la Burochore a tradicione	TD 6
THE POLE	Glassin
Refniere Pouille	15 3.3 Lur
objectif : Voir l'intérêt	Bour la
de l'historique des traitements	Ficher
La Prevairie Les Cannerais	====
	objectif : Voir l'intérêt de l'historique des traitements

Un point très utile à connaitre sur la boite à outils de traitement de QGIS, c'est qu'elle gère un historique des traitements, accessible en cliquant sur 🕓



C'est la liste de toutes les dernières opérations effectuées dans la boite à outils, avec leurs paramètres, en cliquant sur une ligne on voit le détail de l'opération avec ses paramètres, plusieurs mois de traitements sont archivés.

Q Processing History	×
Q. Recherche	
▼ □ Hier	*
[2024-10-16 22:24] Tampon - {'DISSOLVE': False,'DISTANCE': 100,'END_CAP_STYLE': 0,'INPUT': 'U:	
[2024-10-16 22:24] Tampon - {'DISSOLVE': False,'DISTANCE': 250,'END_CAP_STYLE': 0,'INPUT': 'U:	
[2024-10-16 22:23] Tampon - {'DISSOLVE': False,'DISTANCE': 25,'END_CAP_STYLE': 0,'INPUT': 'U:/	
👁 [2024-10-16 17:42] Centroïdes - {'ALL_PARTS': False,'INPUT': 'U:/temp/LA_FLECHE/RPG_2011/RP	
[2024-10-16 15:33] surface_zonale - {'identifiant_zone': 'NOM_COMM', 'parcelles': 'U:/temp/LA_FLE	-
processing.run("native:buffer", -{'INPUT':	*
'U:/temp/LA_FLECHE/RPG_2011/RPG_centro.shp','DISTANCE':100,	
'SEGMENTS':5, 'END_CAP_STYLE':0, 'JOIN_STYLE':0, 'MITER_LIMIT':2,	
'DISSOLVE':False,'SEPARATE DISJOINT':False,'OUTPUT':	•
Clear Save As Fermer Aide	

En double cliquant sur la ligne, on relance la boite de dialogue de l'opération avec les mêmes paramètres.

<b>Q</b> Tampon				Х
Paramètres Journal		Þ	Tampon	•
Couche source U:/temp/LA_FLECHE/RPG_2011/RPG_centro.shp Entité(s) sélectionnée(s) uniquement	- C	• •	Cet algorithme calcule une zone tampon pour toutes les entités d'une couche d'entrée, en utilisant une distance fixe ou dynamique.	
Distance 100.000000 Segments	mètres	•	Le paramètre de segments contrôle le nombre de segments de ligne à utiliser pour approximer un quart de cercle lors de la création de	

Un clic droit sur une ligne permet de faire un copier/coller du code correspondent, dans différents langages :

<b>Q</b> Processing History		×
Q Recherche		
🔻 📄 Hier		<b>A</b>
[2024-10-16 22:24] Tamp	n Copier comme commande Python	': 0,'INPUT': 'U:
[2024-10-16 22:24] Tamp	Copier comme commande ggis process	': 0,'INPUT': 'U:
[2024-10-16 22:23] Tamp	Conjer en tant que ISON	0,'INPUT': 'U:/
@ [2024-10-16 17:42] Cent		
♣ [2024-10-16 15:33] surfs	Créer un Test	/temp/LA_ELE

Exemple de copier / coller en python :



the Blede not le Chaumanism le Gue	SIG
eMauras la Pourillene la Bur ochore stiere a la la Robiniere la Marioire	TD 6
Refniere Pouille	-GIS 3.34 Lure
s <b>Voi</b> r l'intérêt du mode Batch	Bour la la
Roncin L'Aunay Pion	- Children
5	Maupar la Callardere surren la BRobiniere objectif : Voir l'intérêt du mode Batch

Le mode d'exécution « Batch » permet d'exécuter un même outil de traitement de manière automatique sur une série de fichiers d'entrée.







dgur = niveaux d'urbanisation

# 4.1 Le menu « processus de lot » dans la boite à outils de traitements

Découpez en une seule opération les 3 couches «xxx \_europ » par le contour de la France en utilisant le mode « batch »

Utilisez la fonction « Couper » des outils de géotraitement

☆ → Q Recouvrement de vecteur →

Par contre au lieu de la lancer comme d'habitude par un double clique bouton gauche

souris, faire 🖑 sur l'outil « Couper », puis choisir « Exécuter comme processus de lot »

<ul> <li>Recouvrement de vecteur</li> </ul>		
🕐 Couper	Fuéguter	
🔆 Couper avec	Executer	
Différence	Exécuter comme processus de lot	
Différenciati	Éditer les styles de rendu pour les sor	
🎂 Extraire/décor	Iper par etendue	

Vous allez ajouter dans le tableau qui apparait une ligne pour chaque couche à couper par l'outil.

# 4.2. Remplissage du tableau des traitements et nommage automatique

Créer 3 lignes avec le bouton <sup>(1)</sup> et choisir comme couches sources : airport\_eu, dgur\_europ et villes\_europ et comme couche de superposition France\_FR

🔇 Traitement par lots - Couper			
Paramètres Journal	Paramètres Journal		
윤 😑 📄 🔒			
Couche scurce	Couche de superposition	Découpé	
1 Auto-rempl ssage	💂 Auto-remplissage	Auto-remplissage	
2 airport_europ [EPSG:42,8]	🔹 🔧 🛄 🏳 FRANCE_FR [IGNF: 🔹 🔧 具 U:/temp/LA_FLECHE/resultat1.shp	p	
3 🗁 dgur_europ [EPSG:4258]	💌 🔧 🛄 🏳 FRANCE_FR [IGNF: 💌 🔧 具 U:/temp/LA_FLECHE/resultat2.shp	p	
4 ° villes_europ [EPSG:4326]	💌 🔧 📖 🏳 FRANCE_FR [IGNF: 💌 🔧 💷 U:/temp/LA_FLECHE/resultat3.shp	ρ	

#### Découpé : ce sont les fichiers résultats

Cliquez sur \_\_\_\_\_ sur la première ligne de Découpé puis donner un chemin vers un fichier de nom « resultat.shp » comme sortie dans un dossier (ici u:/temp)

Une boite de dialogue apparait, choisir alors « Remplir avec des nombres »

Q Paramètres de remplissage	automatique	×	
Mode de remplissage automatique	Remplir avec des nombres		
Paramètre à utiliser	Couche source	Ŧ	
	ОК	Annuler	

Des noms « automatiques » ont été générés, avec un nombre différent en fin.

	×
Découpé	
_FLECHE_Version_02_2019_2020/LA_FLECHE_WORK/Divers/BATCH/resultat1.shp	
_FLECHE_Version_02_2019_2020/LA_FLECHE_WORK/Divers/BATCH/resultat2.shp	
_FLECHE_Version_02_2019_2020/LA_FLECHE_WORK/Divers/BATCH/resultat3.shp	

Sinon l'alternative est de choisir un nom manuellement pour chaque sortie.

Cochez 🗹 Charger les couches 🛛 en bas à gauche de la boite de dialogue

Exécuter Cliquez finalement sur

On a bien les 3 couches résultats qui sont découpées par le contour de la France.



Pour accélérer les traitements géographiques sur un grand nombre d'objet il est conseillé de préalablement calculer des index spatiaux avec :



Si plusieurs lignes doivent avoir la même valeur, comme FRANCE\_FR pour « couche de superposition », saisir la valeur sur la première ligne, puis cliquer sur « Auto remplissage » en haut de colonne et choisir dans le menu « Remplir » qui va recopier la valeur dans toutes les lignes suivantes (utiles quand on a un grand nombre de lignes).

Couche de superposition	
Auto-remplissage	-
Remplir	
Calculer par expression	

É

Refaire l'opération précédente mais en gardant pour les fichiers résultats un nom automatique qui soit lié au nom de la couche de départ

Répéter comme au point précédent en donnant comme nom de fichier résultat « fr\_.shp », mais cette fois ci choisir comme mode de remplissage automatique :

#### « Remplir avec les valeurs du paramètre » et paramètre « Couche source »

Q Paramètres de remplissage	automatique X
Mode de remplissage automatique	Remplir avec les valeurs du paramètre 💌
Paramètre à utiliser	Couche source
	OK Annuler

Les couches résultats ont alors le même nom que la couche de départ préfixé par «fr\_ »

	Couche source			Couche de superposition			
1	Auto-remplissage		-	Auto-remplissag	е	-	
2	📌 airport_europ [EPSG:4 🔻	2		FRANCE_FR [IGN -	2		U:/temp/LA_FLECHE/fr_airport_europ.shp
3	🟳 dgur_europ [EPSG:425 🕶	ar o		FRANCE_FR [IGN -	2		U:/temp/LA_FLECHE/fr_dgur_europ.shp
4	° villes_europ [EPSG:432 ▼	2		FRANCE_FR [IGN -	e al la		U:/temp/LA_FLECHE/fr_villes_europ.shp

# Reprojetez en mode Batch, les 3 couches résultats en Lambert 93 (EPSG 2154), en créant des fichiers préfixés par « L93\_ »

🜞 🛶 🍳 Outils généraux pour les vecteurs 🛶 🌞 Reprojeter une couche



# À retenir : **Partie 4. Mode Batch de QGIS**



- Le mode Batch permet d'appliquer le même traitement à toute une série de données, Par exemple des centaines de couches dans un répertoire

- Le mode Batch est une fonctionnalité très simple à utiliser

- Avant de penser à faire des scripts penser au mode Batch

5. Le modeleur gra	phique de chaine de l	traitements QGIS	
Mefferration Laurov le Cha	a vande contine of Flette	act le Oraumanissu la Trategne	SIG
Mariane la Genetia J	aladen stander	Robiniere la Bur ochore	<b>10</b> 6
Maillardiere le Plegrit - da		niere Pouille	GIS 3.34
	Durée 30 minutes	🕈 objectif : Maitriser	Bour la
Working Car	le modeleur gro	aphique de traitements	Fichepa
Le et Ala Durandiere	Balancen La Prevair	ta Garrier	

Le modeleur graphique de chaine de traitements QGIS, ou GM (Graphic Modeler) est un outil qui permet de développer des « macros » de séquence de traitements **sans coder**, en utilisant un langage iconique et une interface graphique.

Faire un modèle qui calcule la surface totale de « parcelles » par « zones » définie dans une autre couche : exemple le nombre d'hectares de parcelles agricoles RPG par zone IRIS

Le registre parcellaire graphique (RPG) contient les parcelles agricoles des déclarations PAC des agriculteurs.

Affichez les 2 couches dont on va se servir

LA\_FLECHE → Contours\_Iris → Carto → IRIS\_extrait72 Le champ « Nom\_Iris » donne le nom identifiant de la zone iris LA\_FLECHE → RPG\_2011 → RPG\_2011\_MAJ



13 / 66

jean-marc.gilliot@agroparistech.fr TD SIG 6 Introduction à l'automatisation avec QGIS 3.34

# 5.1. L'interface du modeleur graphique

#### Ouvrir l'interface du modeleur de traitement

**Depuis la boite de traitements**, cliquez sur modèle »

et choisir « Créer un nouveau



#### La fenêtre du Modeleur apparait :

#### Les entrées possibles

#### ou



2 onglets : Choisir affichage des entrées ou des traitements (Algorithmes)

jean-marc.gilliot@agroparistech.fr TD SIG 6 Introduction à l'automatisation avec QGIS 3.34

#### Nommer le modèle

Dans les propriétés du Modèle Nom : surface\_zonale

Propriétés	s du modèle	ð
Nom	surface_zonale	]
Groupe	surface	]
Variable	s Propriétés du modèle	

# Enregistrement du modèle avec 🗐

Pensez à enregistrer régulièrement au cours de la construction du modèle pour ne rien perdre. Regarder dans la boite à outils QGIS un nouvel outil « surface\_zonale » est apparu dans les modèles de la boite à outils de traitements.



#### 5.2 Ajout des données : paramétrisation des entrées

 Ajoutez les 2 entrées de type « couche vecteur » et les nommer « zones » et « parcelles »de notre modèle de traitement

Sur l'onglet « Entrées » à gauche Cliquez sur 🕀 Couche vecteur pour ajouter nos 2 entrées

#### En effet nos entrées sont de type couche SIG vecteur

Entrées	Ø				
🕆 Couche d'annotation	-				
🕀 Couche de maillage			~		~
🕀 Couche générique		🕂 zones	23	🕂 parcelles	23
🕀 Couche nuage de points					
🕀 Couche raster					
순 Couche vecteur					
🕀 Couches de création du maillage triangulaire (TIN)					
🕀 Couches de tuiles vecteur					
🕀 Couches DXF					
문 Couleur	*				

zones et parcelles sont des paramètres d'entrée (variables) ils seront affectés à des données par l'utilisateur au moment de l'exécution, typiquement l'utilisateur choisit dans une liste des couches du projet.

# 5.3 Ajout des algorithmes (traitements)

Intersecter les parcelles par les zones IRIS : croisement de couches
 Recouvrement de vecteur >> Intersection

#### Dans l'onglet « Algorithme » ajouter une opération d'intersection

Algorithmes	Ø		
Q Rechercher			
Différence			- 83
Différence (multiple)	-U- Lonco	-U- parceiles	000
Différence symétrique			
🌞 Extraire/découper par étendue 🛛 🔺			
💮 Intersection			
Intersection (multiple)			
X <sup>™</sup> Intersections de lignes			
P Union			
P Union (multiple)			
Sélection dans un vecteur	<b>*</b>		
Entrées Algorithmes			

Paramétrer l'intersection avec les paramètres d'entrée : parcelles et zones

Q Intersection	
Properties Comments	
Description Intersection	
Show advanced parameters	
Couche source	
Wilisation d'une entrée du modèle	parcelles
Couche de superposition	
Utilisation d'une entrée du modèle	zones

Une boite est apparue sur le diagramme et elle est relié a ses paramètres d'entrée

zones	8	• <del>伊</del> p	oarcelles	***
	• Ent	rée +		
	👘 Int	ersection	83 	
	Sor	tie +		

Par défaut le nom de la « boite » est le nom de la fonctions, mais on peut le changer, pour le rendre plus explicite

Intersection : no1 prétraitement Surtout pour les gros modèles et pour distinguer les étapes qui sont faites avec la même fonction.

#### Testez l'exécution de votre modèle dans le modeleur

Pour faire cela, Changez temporairement le statut de la sortie, on va lui donner un nom plutôt que de les laisser en fichier temporaire (pour qu'il se charge en mémoire).



📟 et Editer

En bas de la boite de dialogue, le champ intersection (même nom que la fonction) est le fichier sorti en résultat, par défaut si vide, c'est un résultat intermédiaire, temporaire.



Entrer un nom pour générer un paramètre résultat final : par exemple « intersect\_out » Intersection



On peut avoir plusieurs paramètres résultats dans un même modèle, si nécessaire.



Testez l'exécution avec 🕨	
Q Modeleur de chaîne de traitement	
📄 🔒 🛃 🅦 🗩 🗩	🖾 🚔 🚣 🏤 🚺 🕨
Propriétés du modèle 🛛 🔊	

L'utilisateur choisit dans les listes les couches à affecter aux paramètres

Q surface_zonale	
Paramètres Journal	
parcelles	1 /
RPG_2011_MAJ [IGNF:LAMB93]	
zones	7
RIS_extrait72 [EPSG:2154]	
intersect_out	2
[Créer une couche temporaire]	<hr/>
✓ Ouvrir le fichier en sortie après l'ex	récution de l'algorithme

P On peut mettre ici un fichier permanent (pas temporaire) si souhaité

Regarder dans le projet QGIS une couche « intersect\_out » a été ajoutée



# 5.4 Le nouveau modèle dans la boite à outils de traitement

Testez l'exécution de votre modèle dans la boite à outils de traitements

🗾 Enregistrer votre modèle et fermer la fenêtre Modeleur

Depuis la boite à outils de traitements de la fenêtre principale QGIS



L'exécution et le résultat sont identiques au point précédent, notre outil est devenu un outil à part entière de la boite de traitements, il pourra même être intégré à d'autres modèles ou scripts.

Ouvrir de nouveau la fenêtre du modeleur par sur « surface\_zonale » dans la boite de traitements et choisir « éditer le modèle »

#### 5.5 Connecter les algorithmes entre eux : input <-> output

Calculer la surface des parcelles intersectées

Supprimer la sortie précédente de l'intersection pour revenir à

Intersection

 [Entrez un nom s'il s'agit d'un résultat final]

Ajouter au modèle une calculatrice de champ:

🔇 Table vecteur → 🌞 Calculatrice de champ

Créer un champ « SURFACE » où on calculera la surface en hectares (\$area / 10000)

On va connecter la sortie de l'intersection (couche temporaire) avec l'entrée de la calculatrice.

Q Calculatrice de	champ		
Properties Co	mments		
Description Calcu	latrice de champ		Clinum and la hautan 🍀
Couche source			Cliquer sur le douton
Wtilisation of	d'une entrée du modèle	parcelles	
Valeur			
Valeur pré-ca	alculée		
✓ Entrée du ma	odèle		Choisir comme type d'entrée :
Sortie d'un a	lgorithme		Sortie d'un algorithme précédent

L'icone du bouton change, montrant qu'on a changé de type de paramètre Et on choisit dans la liste « intersection » créé par l'algo. Intersection



Ce processus est très important car c'est qui va permettre d'enchainer les étapes du modèle les unes à la suite des autres, la sortie de la première étape est ainsi automatiquement connectée sur l'entrée de la seconde etc ...

Dans la suite de la boite de dialogue : Créer un champ « SURFACE » de type Décimal et ajouter une formule pour y calculer la surface des polygones

Nom du	u champ
123	SURFACE
Type d	u champ résultat
123	1.2 Décimal (double)
Longue	eur du champ résultat
123	0
Précisio	on du champ résultat
123	0
Formul	e
	Expression Éditeur de fonction
	C C Re Afficher l'aide
	\$area / 10000 algorithm_id  Intersection_0

Nommer le résultat de la calculatrice « sortie » pour voir le résultat dans QGIS

순 zones	💥	8
	• Entrée +	
	🗊 Intersection 💥	
	Sortie =	
	Intersection: Intersection	
	• Entrée +	
	🕸 Calculatrice de champ	23 
	Sortie + •	
	⇒ sortie	***

Géographiquement le résultat est identique à l'étape d'intersection



NUM_ILOT	049-5027		
CULT_MAJ		Mais si on ouvre la table attributaire	
DepCom	72154		
Nom_Com	FLECHE (LA )		
Iris	0107		
DcomIris	721540107		
Nom_Iris	Zone-Industrielle		
Typ_Iris	A	Un nouveau champ « SURFACE » est app	aru
Origine	1		
SURFACE	1.0187423474211246		

# Récapituler les surfaces précédentes par IRIS

De manière à calculer la surface totale de parcelles pour chaque zone IRIS

Supprimer le paramètre « sortie » de la calculatrice

Ajouter « statistiques par catégorie » à votre modèle

 $\mathbb{Q}$  Analyse vectorielle  $\longrightarrow \Sigma$  Statistiques par catégories

<b>Q</b> Statistiques par catégories
Properties Comments
Description Statistiques par catégories
Couche vectorielle en entrée
Utilisation de la sortie d'un algorithme "Calculé" créé par l'algorithme "Calculatrice de champ"
Champ pour calculer les statistiques (si vide, seul le compte est calculé) [optionnel]
123 SURFACE
Champ(s) avec catégories
123 Nom_Iris
Statistiques par catégorie
sortie

Regarder la table résultat « sortie » qui donne les statistiques de surface par IRIS, en particulier le champ « sum » donne la surface totale de parcelles à moins de 150m d'une rivière, en hectares.

	🔉 stat — Total des entités: 9, Filtrées: 9, Sélectionnées: 0 — 🗆 🗙						
8	/ 🐹 🖶 😂 🐘 🖮 🖂 🖄 🖆 1 🖌 🗮 💫 🧏 🍸 🔳 🌺 🗭 1 166 166 106 106 106 🔍						
	Nom_Iris	count	unique	min	max	range	sum
1	Bousse	48	48	0.04289293876	7.72263796907	7.67974503030	101.941003763
2	Villaines-sous	64	64	0.00447322941	10.9249408264	10.9204675970	168.674796771
3	Verron	19	19	0.00019926011	5.09142646071	5.09122720059	34.0867420699

#### Joindre la table précédente sur les IRIS

Supprimer le paramètre « sortie » de l'étape précédente Ajouter au modèle une opération : « joindre les attributs par valeur de champ »

🔇 Outils généraux pour les vecteurs → 🌞 Joindre les attributs par valeur de champ

# Attention au sens de la jointure.

/!\

🔇 Joindre les attributs par valeur de champ				
Properties Comments				
Description Joindre les attributs par valeur de champ				
Couche source				
Wilisation d'une entrée du modèle         zones				
Champ de la table				
123 Nom_Iris				
Couche en entrée 2				
Utilisation de la sortie d'un algorithme "Statistiques par catégorie" créé par l'algorithme "Statistiques par catégories"				
Champ de la table 2				
123 Nom_Iris				
Couche 2 champs à copier (laissez vide pour copier tous les champs) [optionnel]				
123 sum				
Type de jointure				
123 Prendre uniquement les attributs de la première entité correspondante (un à un)				
Supprimer les enregistrements qui ne peuvent être joints				
123 Non				

Pour éviter d'alourdir la table des Iris on ne demande la jointure que du champ « sum » qui contient la surface totale des parcelles du RPG (par défaut tous les champs seraient copiés)

⊕ zones 🚆 🔶 🔶 ⊕ parcelles	
• Entrée +	
Intersection	
Sortie –	
Intersection: Intersection®	
€ Entrée +	
* Calculatrice de champ	
Sortie + •	
← Entrée +	
∑ Statistiques par catégories 💥 🚥	
Sortie + 🕒	
• Entrée +	
Joindre les attributs par 🗱 valeur de champ 🚥	
Sortie	
sortie 💥 🐭	

Exécuter 🕨 on voit que la sortie a maintenant la même géométrie que les zones Iris



# Regarder les attributs de « sortie » Sum a été importé par la jointure

DepCom	72044
Nom_Com	BOUSSE
Iris	0000
DcomIris	720440000
Nom_Iris	Bousse
Typ_Iris	Z
Origine	NULL
sum	781.1083839535393

Dans la fenêtre QGIS faire une symbologie par dégradés de couleurs sur « sum »



# 5.6 Paramétrer un champ

#### Paramétrer le champ identifiant des zones

Pour pouvoir utiliser ce modèle avec d'autres couches de zone que les IRIS il faut le rendre indépendant du champ identifiant des zones « Nom\_Iris » on va donc **paramétrer ce champ dans le modèle**.

$\bigcirc$ Champs vectoriel Parameter Def $ imes$
Nom du paramètre
identifiant zone
Couche parente
zones
Type de données autorisées
N'importe lequel
Accenter multiples champs

Relier ce champ aux traitements *Statistiques par catégories* et jointure Testez votre modèle avec ce changement

Pour statistiques par catégories changer le type pour « entrée du modèle »

<b>Q</b> Statistiques par catégories				
Properties Comments				
Description Statistiques par catégories				
Couche vectorielle en entrée				
Utilisation de la cortie d'un algorithme "Calculé" créé par l'algorithme "Calculatrice de champ"				
Champ pour calculur les statistiques (si vide, seul le compte est calculé) [optionnel]				
123 SURFICE				
Chample avec catégories				

Faire la même chose pour la jointure

🕀 identifiant zone	🗱 🚛 🖓 🔂	🗱 💮 🖓 parcelles	***
	Entrée 🛨		
	intersection	88	
	Sortie –	Intermetion	
		Intersection	
	letter +	~	
	Calculatrice de	champ 💫	
	Sortie +	•	
	• Entrée	•	
	Statistiques n	ar catégories	
/	Sortie	+ ®	
	€Entrée +		
	Valeur de chamn		
	Sortie + •		
	sortie	\$\$ 000	

Puis choisir « identifiant zone »

26 / 66

### Appliquer votre modèle sur IRIS

Il faut maintenant préciser le champ qui sert d'identifiant pour les zones IRIS

🔇 surface_zonale					
Paramètres Journal					
identifiant zone					
abc Nom_Iris					
parcelles					
RPG_2011_MAJ [IGNF:LAMB93]					
zones					
RIS_extrait72 [EPSG:2154]					
sortie [optionnel]					
[Créer une couche temporaire]					

Appliquer votre modèle sur une autre couche de zone que les IRIS

Afficher « COMMUNES\_RPG » LA\_FLECHE → RPG\_2011 → COMMUNES\_RPG

Ce sont les communes où on a l'information des parcelles RPG



Lancer votre modèle en utilisant COMMUNES\_RPG comme couche de zones

Le champ « NOM\_COMM » contient le nom de la commune et pourra être utilisé comme identifiant de zone.

Surface_zonale     Paramètres   Journal   identifiant zone   abc NOM_COMM   parcelles   Image: Ima				
Paramètres       Journal         identifiant zone       abc NOM_COMM         parcelles       Parcelles         Prog_2011_MAJ [IGNF:LAMB93]				
identifiant zone          abc NOM_COMM         parcelles         Image: RPG_2011_MAJ [IGNF:LAMB93]				
abc NOM_COMM       parcelles       PRG_2011_MAJ [IGNF:LAMB93]				
parcelles PRG_2011_MAJ [IGNF:LAMB93]				
RPG_2011_MAJ [IGNF:LAMB93]				
zones				
COMMUNES_RPG [IGNF:LAMB93]				
sortie [optionnel]				
[Créer une couche temporaire]				







# 6.1. Code dans la console python de QGIS et l'interface iface

On peut accéder à la console python de QGIS avec le bouton 🔛

📿 LA_I	Q LA_FLECHE - QGIS													
Projet	<u>É</u> diter	V <u>u</u> e	<u>C</u> ouche	Préférences	E <u>x</u> tension	<u>V</u> ecteur	<u>R</u> aster	<u>B</u> ase de donné	es <u>I</u> nternet	<u>M</u> aillage	SCP <u>T</u> ra	itement <u>A</u> i	de	
	<u> </u>		🛐 😭	1	ير 🔍 🕏	<b>, 1</b>		R R 🖬	1 🖸 🔁	<b>Q</b>	- 🖳 -	<mark>€</mark> ₀	🏥 🧱 🕺	320
I //_	/ 🕞	°°	/‰ ▼		ð <b>6</b> 4	•	abc 🇌	aba aba	abc abc ab	5 🛛 🔘 (	0. 🙊	2		
0	V <b>.</b> 1.	0.00	4	mètres 🔻	r×x	Ŧ						Console	e Python t+P)	
i 🕵	🍖 V.	P	<b></b>	] 🛛 📴 🔎	• RGB =	•	<u> 🔊 (</u>	🕂 🔎 💽 ROI	<b>I</b>	() Dist 0	.010000	Min 60	Max 100	
Explora	teur					ð 🗙 🛒	Jer 2		050 21		eaumont ur-Sarthe		31	
	7	0					* 26	B. 032	Sillé-le-Guil	-05	+7	Ballon	Bonnét	able

# La fenêtre de console apparaît





On peut saisir du code Python ici les affichages se font ici

Au démarrage de la console est automatiquement exécuté le code suivant (caché), qui intègre la librairie de base de QGIS

```
from qgis.core import *
import qgis.utils
```

# L'interface iface() permet de faciliter l'accès aux objets du projet QGIS.

Or Compter le nombre d'enregistrement de la couche active dans QGIS



Ce code permet de récupérer un pointeur sur la couche active de la table des matières



Ou print(layer.id())

Remarquez que pendant la saisie du code une aide affiche les fonctions en mode auto complétion



Pour compter les enregistrements :

	9	>>> layer.id()
	10	'DEPARTEMENT20170919164816571'
layer.featureCount()		>>> layer.featureCount()
	12	96
	13	

Faire une boucle qui imprime le nom de tous les départements

for fe in layer.getFeatures():
 print(fe.attributes()[2])

Attention à l'indentation en python (décalage en colonne) c'est ce qui marque le début d'un nouveau bloc d'instructions.

Le print() s'affiche dans la fenêtre de console

Console Python 🍾 🌏 I 📄 I 🔧 🔳 305 >>> for fe in layer.getFeatures(): 306 ... print (fe.attributes () [2]) 307 AIN 308 AISNE 309 ALLIER 310 ALPES-DE-HAUTE-PROVENCE 311 HAUTES-ALPES 312 ALPES-MARITIMES 313 ARDECHE 314 ARDENNES 315 ARIEGE 316 AUBE 317 AUDE 318 AVEYRON 319 BOUCHES-DU-RHONE

🗥 Deux fois entrée sur la dernière ligne de code pour lancer l'évaluation

QGIS Python API documentation est sur :

https://qgis.org/pyqgis/3.34/ et aussi https://docs.ggis.org/3.34/en/docs/

# 6.2 Exemples avec iface dans un fichier script de la console

#### 6.2.1 Saisir et exécuter un fichier script dans la console

Dans la fenêtre de la console le bouton 📝 permet d'ouvrir un éditeur de code

Quand on clique sur ce bouton un éditeur apparait

Console Python	🖆 🍃 i 🗟 i 🕨 i 🛰 🖄 🔯 i 🔍 i # 🎢 i 🛓		Ø 🗙
<pre>1 # ·Python ·Console 2 # ·Use ·iface to ·access · QGIS · API · interfac e · or · type · '?' · for ·more · info 3 # · Security ·warning : · typing · commands · fro m · an · untrusted · source · can · harm · your · com puter 4</pre>	Sans titre 0 ★     1     1	/	

Dans un fichier python, faire une boucle qui imprime le nom de tous les IRIS

Fxécuter 🕽 Couche Iris sélectionnée 🕹 🕼 👟 🔭 🖧 🖌 🔌 RPG\_centro\_R RPG centro buffer k la\_fleche villes ROUTE bd\_carthage 
 ✓
 IRIS extrait72

 ●
 DEPARTEMENT
 **5C100 L93** BD\_ALTI LC08\_2013\_07\_10\_L93 Console Python 0 X Bande 4 Bande 3 Bande 2 🍾 🕨 📄 🔧 🔳 🗐 🖆 🍃 i 🗟 🛃 i 🕨 🛰 🗈 🖪 I 🔍 I # 🎢 🗄 🕀 📑 \*Sans titre 0 🗙 1 # · Python · Console • 2 # Use iface to access QGIS API interface or type '? 'for more info 1 layer = qgis.utils.iface.activeLayer() 3 # Security warning: typing commands from an untrust 2-for fe in layer.getFeatures(): ed.source.can.harm.your.computer 4 >>> exec(Path('u:/temp/tmpc6h5enme').read\_text()) 3 print(fe.attributes()[4]) 5 Bousse 6 Clermont-Créans 4 7 Centre-Ville-Sud 8 Centre-Ville-Nord 9 Notre-Dame-des-Vertus 10 Verron 11 Saint-Germain du Val 12 Sainte-Colombe-Monnerie 13 Zone-Industrielle 14 Ligron 15 Villaines-sous-Mal Résultat dans la console 🗹 les noms des Iris le code python

Le 5<sup>e</sup> champ de la table des IRIS, Nom\_Iris, est le nom de l'Iris

Enregistrer le script 🗏

Remarquer le dossier par défaut où sont enregistrés les scripts :

Le plus souvent un dossier dans le profil utilisateur, on peut changer ce dossier dans les réglages de QGIS : Menu > Préférences > Options

Clique sur le bouton « play » 🕨 pour exécuter le script

#### 6.2.2. Accéder aux couches, objets et attributs en python avec iface

On a déjà vu :

- Layer = Iface.activeLayer qui donne la couche active
- Layer.getFeatures() un énumérateur qui renvoie la liste des objets d'une couche que l'on peut traiter dans une boucle For.
- *feature.attributes()* renvoie dans 1 tableau les champs attributaires d'un objet

#### a) Accès aux attributs

Pour accéder à un seul attribut par sa position dans la table ou par son nom :

Feature[2] pour accéder au  $3^e$  champ (premier champ = 0)

Feature[«NOM »] pour accéder à ce même champ par son nom

Faire une boucle qui print le nom des nouvelles régions avec cette syntaxe

#### Par exemple pour les nouvelles régions :

	Champ O	Champ 1	Champ2
NOM_REGION		code_supra	NOUVEAUNOM
1	ILE-DE-FRANCE	11	ILE-DE-FRANCE
2	CENTRE	24	CENTRE
3	BRETAGNE	53	BRETAGNE

```
for fe in layer.getFeatures():
    print( fe["NOUVEAUNOM"] )
```

#### b) Accès aux entités sélectionnées : selectedFeatures()

```
selection = layer.selectedFeatures()
print(len(selection))
for feature in selection:
    # Traitement des entités sélectionnées
```

Faire un script qui calcule et print à la fin, la surface totale de prairies (GRASSLAND) des régions sélectionnées par l'utilisateur : <u>couche REGIONS.SHP</u>

jean-marc.gilliot@agroparistech.fr TD SIG 6 Introduction à l'automatisation avec QGIS 3.34

#### c) Boite de dialogue de type « input text »

Pour demander une valeur à l'utilisateur

from qgis.PyQt.QtWidgets import \*

qid = QInputDialog()
text, ok = QInputDialog.getText(qid, "entrez ", "nom:", QLineEdit.Normal, "par défaut")

print(text)
print(ok)

text contient le texte saisi par l'utilisateur ok vaut True si l'utilisateur a cliqué sur le bouton OK

d). Accès à la géométrie

🔇 entrez	×
nom:	
par défaut	
ОК	Annuler

🗳 Faire un script qui calcule et print la surface (arrondi au km2) des nouvelles régions

Remarquez que dans la table des nouvelles régions il n'y a pas de champ « surface »

La fonction .geometry() renvoie la geometrie d'un objet (feature) La fonction area() renvoie la surface en unités de la couche

```
layer = iface.activeLayer()
for fe in layer.getFeatures():
    geom = fe.geometry()
    print(fe["NOUVEAUNOM"])
    print(round(geom.area() / 1000000) )
```



# e). Calculs dans un champ

Dans les nouvelles régions faire un script qui créé un champ surface et le remplit avec la surface du polygone

```
layer = iface.activeLayer()
layer_provider=layer.dataProvider()
layer_provider.addAttributes([QgsField("SURFACE",QVariant.Double)])
layer.updateFields()
layer.startEditing()
for fe in layer.getFeatures():
    id=fe.id()
    geom = fe.geometry()
    surface = round(geom.area() / 1000000)
    attr_value={3:surface}
    layer_provider.changeAttributeValues({id:attr_value})
```

```
layer.commitChanges()
```

6	Q nouvelles_regions :: Total des entités: 13, filtrées: 13, sélectionnées: 0					
/	/ 🗾 🖶 C   🛱 🖞 🗠 🖻 🖆 🗧 💟 😓 🍸 🗷 🐥 🖓   🖺 🛗 I 🚍 I 🚍 🎕					
at	NOM_REGION 💌	3 =				
	NOM_REGION	code_supra	NOUVEAUNOM	SURFACE		
1	ILE-DE-FRANCE	11	ILE-DE-FRANCE	12063.000000000		
2	CENTRE	24	CENTRE	39472.000000000		
3	BRETAGNE	53	BRETAGNE	27408.00000000		
4	PAYS-DE-LA-LO	52	PAYS-DE-LA-LOIRE	32322.000000000		
5	NORMANDIE	23	NORMANDIE	30115.000000000		
6	CHAMPAGNE-A	31	CHAMPAGNE-ARDENNE-ALS	57690.000000000		

### 6.3. Code processing.run() dans la console python (Toolbox processing)

Dans beaucoup cas on voudra seulement lancer une fonction de la toolbox avec des couches en entrée et en sortie, sans accéder aux données objet par objet (interface iface), la syntaxe **processing.run()** va permettre de faire cela. C'est une alternative assez simple au model builder pour faire un script, avec en plus les capacités de Python.

Faire un script qui extrait dans le canton de La Flèche, les parcelles agricoles en Blé tendre.

Affichez les 2 couches dont on va se servir : dossier : LA\_FLECHE » RPG\_2011 » RPG\_2011\_MAJ dossier : LA\_FLECHE » GEOFLA\_FRANCE » <sup>7</sup>/<sub>0</sub> CANTON.SHP

La méthode : on réalise chaque étape dans QGIS « à la main » et on récupère le code python correspondent depuis l'historique des traitements, pour l'ajouter à notre script.

Ouvrir la console python : 💌 et Afficher l'éditeur de code avec 🗎

On ajoutera le code python ici 🕻







#### Une nouvelle couche du canton de la flèche a été créé :



Pour récupérer le code python correspondent à cette opération, ouvrir l'historique de la



C'est la liste de toutes les dernières opérations effectuées dans la boite à outils, avec leurs paramètres, cliquer bouton droit sur Extraire par expression, puis

#### Copier comme commande Python

Q Historique ×
Aujourd'hui
★ [2023-10-16 11:45] Extraire par expression ★ [2023-10-16 09:50] imputite (CRASS RAST Copier comme commande Python
(2023-10-16 08:48] i.cluster - {'GRASS_REGI Copier en tant que JSON
2023-10-16 08:27] i.maxlik - {'GRASS_RASTER_FORMAT_META': ", 'GRASS_RASTER_FORMAT_OPT': ", '
(2023-10-16 08:26] i.maxlik - {'GRASS_RASTER_FORMAT_META': ", 'GRASS_RASTER_FORMAT_OPT': ", '
2023-10-16 08:25] i.cluster - {'GRASS_REGION_CELLSIZE_PARAMETER': 0.0, 'GRASS_REGION_PARAME
Double-cliquez sur l'objet de l'historique ou collez la commande ci-dessous pour réexécuter l'algorithme
<pre>processing.run("native:extractbyexpression", {'INPUT': 'U:/temp/LA_FLECHE/GEOFLA_FRANCE/CANTON.SHP', 'EXPRESSION':'."NOM_CHF" = \'LA.FLECHE\'.', 'OUTPUT':'TEMPORARY_OUTPUT'})</pre>
Effacer Enregistrer sous Fermer Aide

Coller dans la fenêtre script de la console python :



Remarquer bien la syntaxe 'OUTPUT' : 'TEMPORARY\_OUTPUT' qui dit que le résultat est mis en couche temporaire.

Changer processing.run en processing.runAndLoadResults Run ne charge pas le résultat dans le projet contrairement à runAndLoadResults()

C'est juste à cette étape pour voir le résultat, à la fin quand le script sera terminé, on peut mettre runAndLoadResults que pour les résultats finaux, et en run pour les étapes intermédiaires.

Enregistrer le script 🗏

Clique sur le bouton « play » Pour exécuter le script : A chaque fois que l'on clique sur play, une couche temporaire « Entités correspondantes » est créé



Etape 2 : Extraire les parcelles du canton 🍀 Extraire par localisation

On prend comme relation géographique, par Exemple les parcelles qui intersecte le canton



On récupère le code de cette étape dans l'historique 🕓 pour l'ajouter au script



Pour que les deux étapes s'enchainent bien on va indiquer que la sortie de la première étape est une des entrées de la seconde (en couche temporaire):

Modifier la première ligne avec ce code : canton = processing.runAndLoadsResuts(... La variable canton ainsi créé permet d'accéder aux paramètres du traitement dont par exemple la valeur de la couche de sortie : canton['OUTPUT']

Modifier la seconde ligne pour que le paramètre de couche 'intersect' face référence à la couche canton précédente : 'INTERSECT':canton['OUTPUT']

Remarquer la syntaxe 'nom paramètre' : 'valeur paramètre'

Etape 3 : Extraire les parcelles de blé tendre (CODE RPG = 1) 🌞 Extraire par expression



Comme précédemment faire le lien entre sortie de l'étape précédente et entrée de cette dernière étape.

Basculer aussi les étapes intermédiaires en « run » simple et non runAndLoadResults

÷		Sans titre 0.py 🗶			
	1	canton=proces	sing.run("native:extractbyexpression", { 'INPUT':'U:/temp/LA_FLECHE/GE	OFLA_FRANCE/C	CANTON.
	2	parc=processi	ng.run("native:extractbylocation", {'INPUT':'U:/temp/LA_FLECHE/RPG_20:	11/RPG 2011 M	MAJ.shp
	3	processing.ru	nAndLoadResults("native:extractbyexpression", {'INPUT':parc['OUTPUT']	, 'EXPRESSION'	':'·"CU
	4			,	

Paramétrer le nom du canton dans le script précédent

Pour cela ajouter une variable nom\_canton en début de script pour coder le nom du canton :

÷	P	Sans titre 0.py 🗙
	1	nom_canton='LA FLECHE'
	2	canton=processing.run("native:extractbyexpression", { 'INPUT'
	3	parc=processing.run("native:extractbylocation", { 'INPUT':'U
	4	processing.runAndLoadResults("native:extractbyexpression", · {

Attention à la syntaxe de l'expression de requête dans la seconde ligne, qui doit combiner du texte constant avec la variable nom\_canton :

'EXPRESSION': ' · "NOM\_CHF" = · · \ ' ' · + · nom canton · + · ' \ ' · '

L'opérateur python « + » pour concaténer des chaines de caractère.

Tester en choisissant pour nom\_canton, le canton de PONTVALLAIN



Processing.execAlgorithmDialog() permet d'exécuter une fonction avec sa boite de dialogue et donc que l'utilisateur choisisse les paramètres interactivement.

# Transformer votre paramètre précédent en une liste de contons

÷	2	*Sans titre 0.py 🗙
	1	list_canton=['LA FLECHE', 'PONTVALLAIN', 'MALICORNE-SUR-SARTHE']
	2 -	for nom_canton in list_canton:
	3	<pre>canton=processing.run("native:extractbyexpression", {'INPUT':'U:/tem</pre>
	4	<pre>parc=processing.run("native:extractbylocation", {'INPUT':'U:/temp/LA</pre>
	5	processing.runAndLoadResults("native:extractbyexpression", -{'INPUT':
	6	

<sup>2</sup>Un script consignant les étapes détaillées d'un traitement (ou un modèle de Model Builder) est un outil important de **traçabilité**, c'est-à-dire qui permet de conserver la trace détaillée d'une méthode, ce qui permettra aussi de garantir qu'on appliquera bien la même méthode sur un autre territoire ou à une autre date.

# 6.4. Code dans un script Python dans la boite à outils Traitement : PyQGIS

#### 6.4.1. Création d'un script comme dans la console

reez un nouve	au script p	ython ave	ec le dout	on	ans la do	oite « Trai	Tements	; »
Q *LA_FLECHE - QGIS							- 0	>
Projet Editer Vue Couche Préférer	nces Extension Vecteur R	ister <u>B</u> ase de données	Internet Maillage SCP	Iraitement <u>A</u> ide				
🕒 🗁 🗟 🔝 👫 🛛 🖤 🎍	• 🔍 🕰 🔍 🤤 🗳	P A A 🖥 📲	" 🔁 🔍 🔍 - 🔜	- 🔁 - 📙 📰 🛛	🗄 🎆 🔽 🥅 T 🎾 T	r		
// /    🕆 k • 🖬 🖻 🎙	K 🖻 🖹 👆 🔿 🗰	역 👜 🖷 🤫 🖷		- 🛃 🛛				
💊 🖓 📜 0.00 🗢 mètres 🔹	· Y X X ·							
🤹 🏫 Vî 🔏 🖏 🔯	🔎 🖸 RGB = 🕘 💌 🖌	🔉 🙈 🔎 🖸 ROI 🔰	🏹 🕂 🕓 Dist 0.010000	Min 60 < Ma	🗴 100 🗢 🔎 💽 Aperçu 📑	0 🗘 🖸 🗘 🖨	KML 📕 »	• 🚺
Couches				Hontfort-le-Gesnois		Boîte à outils de	traitements	
≪ @ ≪ ▼ % ▼ ₩ ₩ □	-la-Vivien	1 Weslay-du-maine	Brûlon 1.E501 Alkones	LE MANS	Mondoubleau		🎐 🔧	
✓ la_fleche	t-Aignan	Grez- Gn-Bouere - 025 2	a Sur Sinte Arna	N 10 10 10 10 10 10 10 10 10 10 10 10 10	Calais of Day les Fontaines	Morée II Créer un nouve	au script	
ROUTE	-sur-Rob // Dzo	Bierné Bierné	SARTH	B 20 052 He Grand-	Stor Savigny- Sur-Braye	Creer un nouve	au script depuis un mod	Jele
bd_carthage		R sur-Sittle	Malicome Ecommo		A S S S S S S S S S S S S S S S S S S S	Loques 8 Ouvrir un script	existant	
✓ □ DEPARTEMENT	0778		Pontvallain			Ajouter un scrip	of vers la bolte a outils	
oans la fenêtre	e Sauver	Exécut	er					
🔇 essai pv -	diteur des	cripts de t	traitement					

	<b>P</b>	🛛 🛃 🕨 🔫 👔 📋 🤚 🍖 🔍 A+ A-
	1	from qgis.core import *
L	2	from qgis.utils import *
	3	import processing

#### 6.4.2. Les « import » python en début du script

Pour pouvoir utiliser certains fonctions internes de QGIS ils faudra charger en début de script certaines librairies, par exemple les liens minimums sont:



#### 6.4.3. processing.run() : les algorithmes de la boite de Traitement dans un script :

Le plus souvent, pour les scripts les plus simples on ne ferra qu'enchainer des fonctions de la boite à outils de traitement.

Créer un script qui appelle la fonction centroïde de QGIS sur la couche sélectionnée



#### 6.4.4. « décorateur » @alg : créer un script dans la boite de traitements :

Faire une nouvelle version du script précédent dans la boite à outil de traitement à l'aide de la syntaxe de décorateur @alg



# Ajouter ce script dans la boite à outils de Traitements avec le bouton 🔍 :



Dans la boite à outils on voit sous « Python » notre script dans sa catégorie

- 🕨 🛞 SAGA
- 🔻 🥐 Scripts
  - TD scripts
    - 🥐 test centro

#### Un script intégrer dans la boite à outils aura une boite de dialogue associée

import alg pour importer la syntaxe @alg

@alg(name='test\_centro', label='test centro',group='TDscripts', group\_label='TD scripts') Défini le nom du script et le nom du groupe où il est placé dans la toolbox

@alg.input pour les paramètres d'entrée / sortie

Cliquer sur le script pour le lancer :

Une boite de dialogue est automatiquement créée, comme pour toutes les fonctions internes de QGIS. On peut choisir les couches en entrée et en sortie.

er carcano	~
Paramètres Journal	test centro
Input layer  DEPARTEMENT [EPSG:2154]  Entité(s) sélectionnée(s) uniquement vecteur output  Enregistrer dans un fichier temporaire]  Ouvrir le fichier en sortie après l'exécution de l'algorithme	Description of the algorithm. (If there is no comment here, you will get an error)
0%	Annuler

Attention des erreurs de syntaxe dans le script peuvent bloquer son importation dans la toolbox, voir même planter QGIS, par exemple le commentaire dans la définition de la fonction (def) est obligatoire.

Les scripts importés dans la ToolBox sont copiés dans ce dossier, dans votre profil : jmgilliot > AppData > Roaming > QGIS > QGIS3 > profiles > default > processing > scripts C'est ce fichier qui est lié à la toolbox maintenant.

#### Afficher des messages dans la boite de dialogue du script

L'objet « feedback » possède des méthodes d'affichage de messages telle que feedback.pushInfo

Ajouter des messages dans le script :

```
23
24 feedback.pushInfo("***** information *****")
25
26 return { 'OUTPUT': res['OUTPUT'] }
27
28
```

Regarder dans la boite de dialogue, après exécution du script :

#### Dans le volet « Journal »



#### 6.4.5. Script R dans la Toolbox de traitement

On peut aussi intégrer des scripts en langage R dans la Toolbox de traitement.

Vérifier que l'extension 🖙 Processing R Provider est bien installée.

On peut ajouter un script R en cliquant ici dans la Toolbox Traitement





Faire un script R qui affiche les statistiques de base d'un champ d'une couche.

<b>Q</b> stat_champ.rsx - R Script Editor							
	🗐 🔜   🕨   🛰 🖨 📔   🗧						
1	##Stat champ=name						
2	##scripts R=group						
3	##couche=vector						
4	##champ=Field couche						
5							
e							
7	<pre>summary(couche[[champ]])</pre>						
я							

## est le symbole de « décoration » pour entrer les paramètres du script. Nom de variable=propriété

Stat champ=name est le nom du script

Group est le groupe de scripts

Couche= vecteur la variable couche reçoit une couche de type vecteur Champ est une variable pour un champ de la couche précédente

🔇 Stat champ								
Paramètres Journal								
couche								
DEPARTEMENT [EPSG:2154]	C DEPARTEMENT [EPSG:2154]							
Entité(s) sélectionnée(s) uniquement	Entité(s) sélectionnée(s) uniquement							
champ								
123 POP2016								
R Console Output [optionnel]								
[Enregistrer dans un fichier temporaire]								

Les résultats dans le Journal

```
🔇 Stat champ
```

ĺ	Paramètres Journal
	<pre>[1] "C:/Users/jmgilliot/AppData/Roaming/QGIS/QGIS3/profiles/default sf"</pre>
	<pre>Linking to GEOS 3.9.1, GDAL 3.3.2, PROJ 7.2.1; sf_use_s2() is TRUE [1] "C:/Users/jmgilliot/AppData/Roaming/QGIS/QGIS3/profiles/defaul1 raster" La shargement a pl@cossitle la package : sp</pre>
	Min. 1st Qu. Median Mean 3rd Qu. Max. 75784 301306 537809 672965 847263 2617319

Ou en cliquant sur le lien dans le volet

Visualiseur de Résultats

R Console Output [04:27:44PM]

R Console Output [04:13:44PM]

R Console Output [04:10:15PM]

# **R** Output

```
[1] "C:/Users/jmgilliot/AppData/Roaming/QGIS/QGIS3/profiles/default/processing/rlibs/sf"
Linking to GEOS 3.9.1, GDAL 3.3.2, PROJ 7.2.1; sf_use_s2() is TRUE
[1] "C:/Users/jmgilliot/AppData/Roaming/QGIS/QGIS3/profiles/default/processing/rlibs/raster"
Le chargement a nÃ@cessitÃ@ le package : sp
Min. 1st Qu. Median Mean 3rd Qu. Max.
75784 301306 537809 672965 847263 2617319
```



# À retenir :

# Partie 6. Script python et R dans l'interface de QGIS



- On peut saisir et exécuter du code python à différents endroits dans l'interface QGIS :

- Directement dans la console python
- Dans l'éditeur de code de la console python
- Intégrer son code comme un nouvel outil dans la boite de traitement

- La façon le plus simple d'utiliser les opérations internes de QGIS est par la fonction processing.run(), mais ne permet pas de manipuler le projet et les objets géographiques (polygones par ex) individuellements.

- L'interface iface permet quant à elle cette manipulation du projet et des objets individuellement

- La syntaxe décorateur @alg permet de faire des scripts intégrés dans la boite à outils avec une boite de dialogue.

 L'extension Processing R provider permet d'ajouter des scripts R dans la boite à outils, on peut ainsi accéder en langage R aux objets du projets et utiliser tous les traitements R.



# 7.1. ggis\_process : Code « processing » dans un script externe à QGIS

Comme dans le cas précédent pour faire des scripts n'utilisant que des appels à des fonctions de la boite à outils de traitement, mais cette fois dans un script externe à QGIS, sans que QGIS soit ouvert.

Depuis quelques versions de QGIS a été introduit **qgis\_process** un exécutable dans l'installation de QGIS, qui permet directement depuis la ligne de commande ou depuis un fichier script, de lancer des fonctions de la boite à outils de traitements, sans avoir à lancer QGIS lui-même. Utilisable potentiellement depuis n'importe quel langage qui accepte des appels systèmes (pour appeler qgis\_process) : batch windows (.cmd .bat), script shell Linux (.sh), script python, script R, visual basic dans Excel etc ...

Consulter les explications détaillées sur le site de QGIS : https://docs.ggis.org/3.34/fr/docs/user\_manual/processing/standalone.html

Sous Windows qgis\_process.exe est localisé dans le sous dossier QGIS : apps/qgis-ltr/bin ou apps/qgis/bin

Sous windows on n'appelle généralement pas qgis\_process.exe directement, mais un fichier de commande .bat qui initialise aussi les variables QGIS, depuis le sous-dossier QGIS : Bin/qgis\_process-qgis-ltr.bat ou bin/ qgis\_process-qgis.bat Ouvrir une fenêtre de commande (ou powershell) windows



Cela ouvre une fenêtre de commande

Ajouter le chemin de QGIS à la variable windows PATH



Usage: C:\PROGRA~1\QGIS 3.34.4\apps\qgis-ltr\bin\qgis\_process.exe [--he command] [algorithm id, path to model file, or path to Python script] [p

Les commandes QGIS sont maintenant accessibles en ligne de commandes.

Sous Linux qgis\_process est normalement directement accessible, après installation.

#### 7.1.1 qgis\_process en ligne de commande

Créer une nouvelle couche des centroïdes des parcelles RPG avec une ligne de commande utilisant ggis\_process

LA\_FLECHE >>> RPG\_2011 >>> RPG\_2011\_MAJ

Repérer le chemin complet sur le disque des données : par exemple : U:/temp/LA\_FLECHE/RPG\_2011 taper cette commande dans la fenêtre de commande (shell) + 📛

**qgis\_process-qgis-ltr run native:centroids --**INPUT=U:/temp/LA\_FLECHE/RPG\_2011/RPG\_2011\_MAJ.shp --OUTPUT=U:/temp/LA\_FLECHE/RPG\_2011/RPG\_centro.shp

#### (run pour exécuter une commande)

Invite de commandes	- 🗆	×
C:\Users\jmgilliot>qgis_process-qgis-ltr run native:centroidsINPUT=U:/temp/LA_FLECHE/RPG_2011/RPG_2011_ OUTPUT=U:/temp/LA_FLECHE/RPG_2011/RPG_centro.shp	MAJ.shp	î
Inputs		
INPUT: U:/temp/LA_FLECHE/RPG_2011/RPG_2011_MAJ.shp OUTPUT: U:/temp/LA_FLECHE/RPG_2011/RPG_centro.shp		
0102030405060708090		
Results		
OUTPUT: U:/temp/LA_FLECHE/RPG_2011/RPG_centro.shp		
C:\Users\jmgilliot>		

Dans le panneau explorateur de fichier sous QGIS le fichier résultat est apparu :



#### Ajouter le dans le projet QGIS



Renouveler le traitement mais en se plaçant dans le dossier projet pour avoir des chemins plus courts vers les données

Dans la fenêtre de commande utiliser l'instruction CD (change directory) pour se mettre dans le dossier de LA\_FLECHE

U: pour changer de disque si nécessaire Puis cd /temp/LA\_FLECHE



Relancer la commande « qgis\_process » précédente avec des chemins relatifs aux données, cela fait des chemins moins long à entrer.



On peut aussi ouvrir une fenêtre de commande à partir de l'explorateur de fichiers Windows, directement dans le dossier LA\_FLECHE



python « standalone »

Touche Shift + clic droit souris sur le dossier LA\_FLECHE



On est dans le bon dossier directement

Choisir Ouvrir la fenêtre ici 7.1.2 ggis\_process dans un script

# Renouveler le traitement précédent mais depuis un script python

Créer à la racine de LA\_FLECHE un fichier texte « script.py » et l'éditer avec un IDE python comme **spyder (pour python 3)** par exemple



Subprocess est une librairie on y trouve la fonction run qui permet d'exécuter une commande externe à Python, il existe plusieurs fonctions pouvant le faire en python

Cmd est une variable chaine de caractère dans laquelle on prépare la commande à exécuter, remarquer l'antislash « \ » en fin de ligne qui permet d'introduire un saut de ligne pour une meilleur lisibilité du code.

Exécuter le code avec 📰 le résultat est nommé « RPG\_centro\_python.shp »

Ce résultat apparait dans le dossier RPG\_2011:



Calculer un buffer de 100m à partir des centroïdes précédent, en passant par un fichier temporaire de centroïde et en enchainant les deux opérations dans le même script python.



La librairie tempfile permet de créer un dossier (ou fichier) temporaire pour le fichier intermédiaire des centroïdes.



On peut récupérer dans une variable des informations, sur la réussite de la fonction et autres messages d'erreur avec une syntaxe du type : res = subprocess.run()

jean-marc.gilliot@agroparistech.fr TD SIG 6 Introduction à l'automatisation avec QGIS 3.34

7.1.3 ggis\_process dans un script R « standalone »

Créer un fichier script.R à la racine du projet et Ouvrir RStudio



Créer une nouvelle couche des centroïdes des parcelles RPG avec une ligne de commande utilisant ggis\_process sous R

On pourrait aussi sous R, faire un appel system à qgis\_process comme dans les exemples python, mais on va ici utiliser **le package R « qgisprocess »** qui facilite le travail.

Installer le package avec Install Packages

RStudio File Edit Code View Plots Session Build Debug Profile	Tools Help
💽 🗸 🥘 🚭 🖌 🔚 拱 🥌 📝 Go to file/function	Install Packages
Console Terminal × Background Jobs ×	Check for Package Updates
Install Packages Install from:	

Ouvrir le fichier Script.R

<u>Fixer le répertoire courant de RStudio au dossier du script :</u> Menu : Session >> Set Working Directory >> to Source File Location

Mena · Dession · · Der Working Directory · · To Dource rife Education

Saisir le code et cliquer sur 🕒 Source pour exécuter tout le script

Scr	pt.R ×
	🔊 📄 Source on Save 🔍 🎢 🗸 📋
1	library("qgisprocess")
2	
3	results <- qgis_run_algorithm("native:centroids",
4	<pre>INPUT="./RPG_2011/RPG_2011_MAJ.shp",</pre>
5	OUTPUT="./RPG_2011/RPG_centro_R.shp"
6	)
7	

Le fichier RPG\_centro\_R a bien été créé



Calculer un buffer de 100m à partir des centroïdes précédent, en passant par un fichier temporaire de centroïde et en enchainant les deux opérations dans le script R.



Remarquer la fonction tempfile() qui permet de créer un nom de fichier temporaire. Le fichier RPG\_centro\_buffer\_R.shp a bien été créé





Il y a de nombreux packages R pour le traitement des données géographiques, que l'on peut utiliser en complément de ggisprocess, on peut citer :

- Package « sf » (simple feature) pour les données vectorielles
- Package « terra » (anciennement raster) pour le traitement des rasters

On peut trouver dans ces packages des fonctions de traitements géographiques équivalentes à celles de QGIS, par exemple dans le package « sf » une fonction de zone buffer : Buffer <- st\_buffer(points, dist=200)

# 7.2 programmes python externe à QGIS

Il est possible de programmer une application externe en python, utilisant toutes les fonctions de QGIS, un peu plus compliquer à faire.



### 8.1 Notion de fichier Markdown

Markdown est un langage permettant d'introduire des balises de mise en forme très simple, en format texte, à l'aide de symboles prédéfinis, un peu à l'image des balises HTML.

Par exemple dans NotePad++ avec le plugin markdown panel



#### 8.2 Fichier RMarkdown

RMarkdown est un format de fichier basé sur Markdown, mais permet beaucoup plus de chose, il a été créé pour R, il permet de générer des rapports dynamiques, mélangeant du texte mis en forme, du code informatique, et des graphiques produits par ce code.

L'extension de fichier pour RMarkdown est «.Rmd», le rapport qui est généré dynamiquement est généralement en format : html, Pdf ou Word.



Vérifier que le package rmarkdown et tinyTex (LaTex) est bien installé

#### Créer un fichier R Markdown dans RStudio :

RStudio									
File E	dit C	Ode	View	Plots	Session	Build	Deb	ug Profile Tools	Help
New File 🕨							R Script	Ctrl+Maj+N	
New Project							Quarto Document		
Open File Ctrl+O						Quarto Presentatio	on		
Open File in New Column							R Notebook		
Reopen with Encoding						P. Markdown			
Recent Files							K Warkoown		

#### On choisit un type « Document » format de sortie en PDF par exemple

New R Markdown							
Document	Title:	test_Rmarkdown					
🛱 Presentation	Author:						
🛞 Shiny	Date:	2024-10-17					
📙 From Template	Use current date when rendering document						
	Default Output Format:						
	O HTML						
	Recommended format for authoring (you can switch to PDF or Word output anytime).						
	PDF						
	PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).						
	<u> </u>						

Enregistrer dans un fichier .rmd avec 🔚



Le fichier mélange du texte, éventuellement mis en forme et des parties nommées « chunk » qui contiennent du code ici en R.

Faire un fichier RMarkdown, qui compte le nombre de parcelles RPG dans les ilots IRIS, en utilisant ggisprocess et qui fait en graphique R pour présenter les résultats.

Utiliser :

- « native :centroids » pour convertir les parcelles en points
- « native: countpoints inpolygon » pour compter le nombre de points par polygone



```
1 ----
 2 title: "test_Rmarkdown"
 3 output: word_document
 4 date: "2024-10-17"
 5 - -
6
7 - ``{r setup, include=FALSE,echo=FALSE}
8
9
   library("qgisprocess")
10 library("sf")
11 library("ggplot2")
12
13 -
14
15 - ## Traitements QGIS pour le comptage de parcelles
16
17 Dans cette partie traitement géographique avec QGIS comptage des *parcelles RPG* par zone IRIS
18
19 - [``{r comptage, warning=FALSE, message=FALSE}
                                                                                               谷 🎽 🕨
20
21
22
23 centroids <- tempfile(fileext = ".shp")</pre>
24
25 results <- qgis_run_algorithm("native:centroids",</pre>
26
                                   INPUT="RPG_2011/RPG_2011_MAJ.shp",
27
                                   OUTPUT=centroids
   )
28
29
30 results <- qgis_run_algorithm("native:countpointsinpolygon",</pre>
31
                                   POINTS=centroids,
                                   POLYGONS="Contours_Iris/carto/IRIS_extrait72.shp".
32
33
                                   OUTPUT="RPG_2011/comptage.shp"
34
   )
35
36 - . . .
37
38 - ## Graphique des résultats
39
40 Le graphique ggplot du nombre de parcelles comptées par ilot IRIS
41
42 - ``{r graphique}
                                                                                                승 🗶 🕨
43
44 comptage <- read_sf("RPG_2011/comptage.shp")</pre>
45
46 ggplot(comptage,aes(x=Nom_Iris,y=NUMPOINTS))+geom_bar(stat="identity")+
      theme(axis.text.x = element_text(angle = 30, hjust = 1, vjust = 1))+ggtitle("Nombre de
47
    parcelles RPG par ilot IRIS")
48
49
50 -
```

En installant le package R « Reticulate » on peut aussi insérer des chunk en python dans son RMD, on peut même partager les objets entre python et R

Après exportation par <u>*Knit*</u> le fichier résultat est ci-dessous :

# test\_Rmarkdown

2024-10-17

#### Traitements QGIS pour le comptage de parcelles

Dans cette partie traitement géographique avec QGIS comptage des *parcelles RPG* par zone IRIS

#### Graphique des résultats

Le graphique ggplot du nombre de parcelles comptées par ilot IRIS

```
comptage <- read_sf("RPG_2011/comptage.shp")</pre>
```

```
ggplot(comptage,aes(x=Nom_Iris,y=NUMPOINTS))+geom_bar(stat="identity")+
    theme(axis.text.x = element_text(angle = 30, hjust = 1, vjust = 1))+ggtitle("Nom
    bre de parcelles RPG par ilot IRIS")
```



#### Nombre de parcelles RPG par ilot IRIS

```
64 / 66
```

1. Démarrage de QGIS	3
2. Chaine de traitements et traçabilité	5
3. Historique de la boite à outils de traitement de QGIS	6
4. Mode Batch (processus de lot) de QGIS	8
4.1 Le menu « processus de lot » dans la boite à outils de traitements	9
4.2. Remplissage du tableau des traitements et nommage automatique	9
5. Le modeleur graphique de chaine de traitements QGIS	13
5.1. L'interface du modeleur graphique	14
5.2 Ajout des données : paramétrisation des entrées	15
5.3 Ajout des algorithmes (traitements)	16
5.4 Le nouveau modèle dans la boite à outils de traitement	19
5.5 Connecter les algorithmes entre eux : input <-> output	19
5.6 Paramétrer un champ	25
6. Script python et R dans l'interface de QGIS	30
6.1. Code dans la console python de QGIS et l'interface iface	30
6.2 Exemples avec iface dans un fichier script de la console	34
6.2.1 Saisir et exécuter un fichier script dans la console	34
6.2.2. Accéder aux couches, objets et attributs en python avec iface	35
a) Accès aux attributs	35
<ul> <li>b) Accès aux entités sélectionnées : selectedFeatures()</li> </ul>	35
c) Boite de dialogue de type « input text »	36
d). Accès à la géométrie	36
e). Calculs dans un champ	37
6.3. Code processing.run() dans la console python (Toolbox processing)	38
6.4. Code dans un script Python dans la boite à outils Traitement : PyQGIS	44
6.4.1. Création d'un script comme dans la console	44
6.4.2. Les « import » python en début du script	44
6.4.3. processing.run() : les algorithmes de la boite de Traitement dans un script	:.44
6.4.4. « décorateur » @alg : créer un script dans la boite de traitements :	45
6.4.5. Script R dans la Toolbox de traitement	47
7. Script python et R « standalone » en dehors de QGIS	51
7.1. qgis_process : Code « processing » dans un script externe à QGIS	51
7.1.1 qgis_process en ligne de commande	52
7.1.2 qgis_process dans un script python « standalone »	54
7.1.3 qgis_process dans un script R « standalone »	57
7.2 programmes python externe à QGIS	59
8. Script python et R « standalone » en RMarkdown	60
8.1 Notion de fichier Markdown	60
8.2 Fichier RMarkdown	61